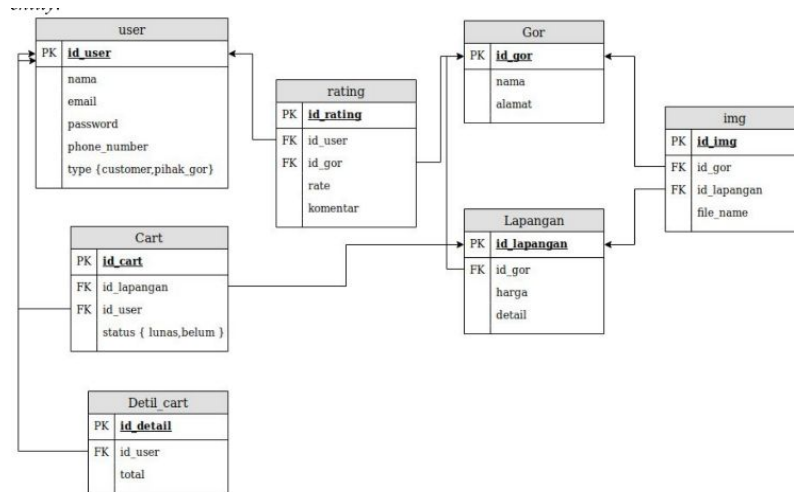


Anggota Kelompok:

- 1301170399 - Adzkar Fauzie Rahman
- 1301170379 - Ghozy Ghulamul Afif
- 1301174008 - Hafizh Jihaad Husni
- 1301174224 - Muhammad Ridaffa Purnomo

Automation Testing



Disini saya menggunakan laravel untuk aplikasi ini. Saya akan melakukan automation testing dengan mengetest tiga kelas, yaitu class User, Rating, dan Gor. Hal pertama yang dilakukan adalah dengan membuat factory dari ketiga table tersebut sesuai dengan model yang sudah dibuat. Saya membuat factory filenya menjadi seperti berikut:

#1 User Factory

```
<?php

/** @var \Illuminate\Database\Eloquent\Factory $factory */
use App\User;
use Illuminate\Support\Str;
use Faker\Generator as Faker;
use Carbon\Carbon;

/*
|-----
|-----
| Model Factories
|-----
|-----
|
| This directory should contain each of the model factory
definitions for
```

```

| your application. Factories provide a convenient way to generate
new
| model instances for testing / seeding your application's
database.
|
*/

$factory->define(User::class, function (Faker $faker) {
    return [
        'name' => $faker->name(),
        'email' => $faker->unique()->safeEmail,
        'email_verified_at' => now(),
        'password' =>
'$2y$10$92IXUNpkj00r0Q5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi',
        'remember_token' => Str::random(10),
        'phone_number' => $faker->phoneNumber(),
        'type' => $faker->randomElement(['customer', 'pihak_gor'])
    ];
});

```

#2 Rating Factory

```

<?php

/** @var \Illuminate\Database\Eloquent\Factory $factory */
use App\Rating;
use Illuminate\Support\Str;
use Faker\Generator as Faker;
use Carbon\Carbon;

/*
|-----
|-----
| Model Factories
|-----
|-----
|
| This directory should contain each of the model factory
definitions for
| your application. Factories provide a convenient way to generate
new
| model instances for testing / seeding your application's

```

```

database.
/
*/

$factory->define(Rating::class, function (Faker $faker) {
    return [
        'rate' => random_int(0,5),
        'komentar' => $faker->sentence(),
        'user_id' => random_int(1, 10),
        'gor_id' => random_int(0, 10)
    ];
});

```

#3 Gor Factory

```

<?php

/** @var \Illuminate\Database\Eloquent\Factory $factory */
use App\Gor;
use Illuminate\Support\Str;
use Faker\Generator as Faker;
use Carbon\Carbon;

/*
|-----
|-----
| Model Factories
|-----
|-----
|
| This directory should contain each of the model factory
| definitions for
| your application. Factories provide a convenient way to generate
| new
| model instances for testing / seeding your application's
| database.
|
|*/

$factory->define(Gor::class, function (Faker $faker) {
    return [
        'nama' => $faker->sentence(),

```

```
        'alamat' => $faker->address()  
    ];  
});
```

Lalu saya membuat file testing dengan menggunakan perintah berikut:

```
$ php artisan make:test UserRatingGorTest
```

Isi file UserRatingGorTest dengan kode berikut:

```
<?php  
  
namespace Tests\Feature;  
  
use App\User;  
use App\Gor;  
use App\Rating;  
  
use Tests\TestCase;  
use Illuminate\Foundation\Testing\WithFaker;  
use Illuminate\Foundation\Testing\RefreshDatabase;  
  
class UserRatingGorTest extends TestCase  
{  
    /**  
     * A basic feature test example.  
     *  
     * @return void  
     */  
    public function testExample()  
    {  
        $users      = factory(User::class, 10)->create();  
        $userCount = count($users) == 10;  
        $this->assertTrue($userCount);  
  
        $gors       = factory(Gor::class, 10)->create();  
        $gorCount  = count($gors) == 10;  
        $this->assertTrue($gorCount);  
  
        $ratings    = factory(Rating::class, 10)->create();  
        $ratingCount = count($ratings) == 10;  
        $this->assertTrue($ratingCount);  
    }  
}
```

```
}  
}
```

Running testing dengan perintah berikut:

```
$ ./vendor/bin/phpunit tests/Feature/UserRatingGorTest.php
```

Hasil dari running akan menjadi seperti gambar berikut:

```
./vendor/bin/phpunit tests/Feature/UserRatingGorTest.php  
PHPUnit 7.5.15 by Sebastian Bergmann and contributors.  
  
.  
1 / 1 (100%)  
  
Time: 731 ms, Memory: 18.00 MB  
OK (1 test, 3 assertions)
```